



# Load Balancing/High Availability Configuration for neonInsight Server

## ***Introduction:***

This document provides details on the configuration and the components used for a two node load balancing system with the neonInsight server. This white paper provides a workable solution to customers who want to deploy a fault-tolerant system with load-balancing at minimal cost and complexity. Below are the steps to create the configuration.

## ***Software used:***

Debian (Etch and Lenny distributions) GNU/Linux: <http://www.debian.org>  
"balance" (<http://www.inlab.de/balance.html>): available in Debian Lenny distribution package

Balance is a load balancing solution; it is a simple but powerful generic TCP proxy with round robin load balancing and failover mechanisms.

nfs-kernel-server  
nfs-common  
neonInsight 5.0-10506: <http://www.bynari.net/public/products/neonInsight/latest/stable>

## ***Definitions:***

The "neon A" server is running Debian Lenny, neonInsight Server, and the balance software. All mail request activity is routed to this server. The neonInsight Server installation is only for the convenience of LDAP installation and maintenance. All other processes of neonInsight Server are not needed and should be stopped. This scenario could be further simplified by installing a standalone LDAP server and importing the data from a running neonInsight Server. For the purpose of this document, this server also is simulating a SAN or some other network attached storage by exporting an NFS share to the other servers. The use of the nfs-kernel-server package is encouraged over the nfs-user-server.

The "neon B1" and "neon B2" servers are running either Debian Etch or Lenny and neonInsight Server. As email traffic passes through neon A, it is forwarded (email proxy) onto neon B1 and neon B2. neon B1 and neon B2 are configured to use a remote LDAP server and remote storage for mail.

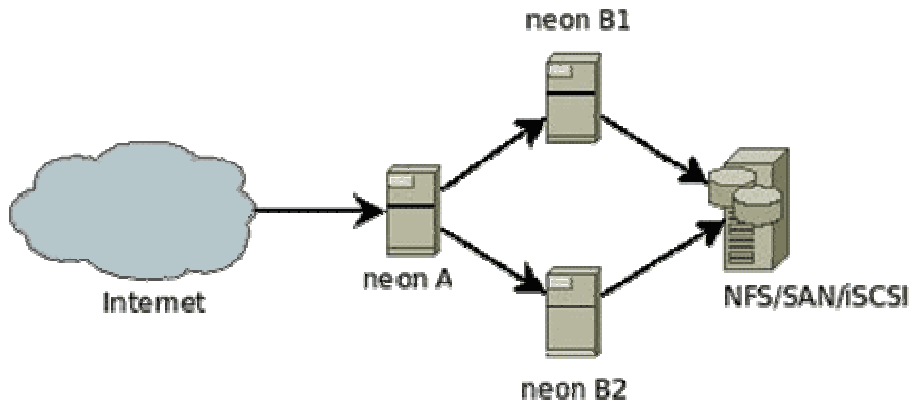


Figure 1 – Load Balanced Network with HA Failover

### Considerations:

When configuring NFS, it is important to have lockd and statd running. Installing the nfs-kernel-server provides the “lockd” and “statd” services.

To verify that lockd is running, type the following command:

```
ps ax | grep lockd <enter>  
11133 ? S 0:00 [lockd]
```

This output shows that “lockd” is running.

To verify that statd is running, type the following command:

```
ps ax | grep rpc <enter>  
11451 ? Ss 0:00 /sbin/rpc.statd
```

This output shows that “statd” is running.

The nfs-common package also provides these processes. Make sure that Neon A, neon B1, and neon B2 are running both lockd and statd. Without them, neonInsight Server will not be able to obtain a proper lock on the shared mail storage files.

balance is available as a Debian package in the Lenny or Sid distributions but not in Etch. It is also available in source code form from the 'balance' website (<http://www.inlab.de/balance.html>).

The test environment in this paper is intended to demonstrate how load balancing and failover function. It is not intended to be a “Best Practices” configuration. In this example, there are three single failure points (OpenLDAP, Balance, and the NFS store). In a real-world environment, redundancy is necessary for each of these points. It is also worth noting that each networked environment is unique. It is not possible to address every kind of failover scenario in this article.



## Configuration:

### neon B1 and neon B2

1. Install neonInsight server on neon B1 and neon B2.
2. Enter the neonInsight change root shell by typing:  
`neonInsight shell <enter>`
3. The user ID (UID) and group ID (GID) of the "neon" user (system user created during the neonInsight server installation process) will be needed when configuring neon A. Enter the following command to display the UID and group ID (GID) of the neon user.  
`id neon <enter>`  
Results should resemble the following:  
`uid=102(neon) gid=104(neon) groups=104(neon)`

**NOTE:** Make note of the UID and GID for use in the next section.

### neon A

1. Install neonInsight server on neon A.
2. Log into neon A and configure some neonInsight user accounts for testing.
3. Stop neon, ejabberd, apache2, amavis, clamav, spamassassin, and all other neonInsight server services except LDAP.
  - a. To stop the individual services, follow the below steps:
    - i. Log into the neonInsight shell  
`neonInsight shell <enter>`
    - ii. Change directories to the /etc/init.d directory:  
`cd /etc/init.d <enter>`
    - iii. Run the list command to see a list of all the services running under init.d:  
`ls <enter>`
    - iv. Among the services listed is apache2. To stop the apache2 service (or any of the neon services), type the following command:  
`/etc/init.d/apache2 stop <enter>`
    - v. Repeat step iv for each service that is to be stopped. The only neonInsight service that needs to continue running is slapd.
4. Configure the NFS server on neon A. Allow neon B1 and neon B2 to access the share as in this example:  
(Using vi or the text editor of your choice)  
`vi /etc/exports:`  
`/mnt/exports 192.168.81.130(rw) 192.168.81.131(rw)`
5. Create a user named neon (system user) on the NFS server (neon A). Make sure that the UID is the same as the neon user account on the neon B1 and neon B2 servers.

**NOTE:** In this example, neon A is serving multiple roles. One of those roles is as an NFS server. The neon user account is configured on neon A for the purpose of granting write access from neon B1 and neon B2 to the shared storage.

6. Create a group named neon (system group) on the NFS server (neon A). Make sure that the GID is the same as the neon group on the neon B1 and neon B2 servers.



## NFS

1. Mount the NFS share on neon B1 and neon B2 under the neonInsight Server change root. Test read and write access as the neon user account created in step 5 of the previous section. **(Configuration – neon A)**  
On the server command line type:  

```
# neonInsight shell <enter>
# mkdir /mnt/neon <enter>
# chown neon.neon /mnt/neon <enter>
# mount -t nfs neonA.example.net:/mnt/exports /mnt/neon <enter>
# su neon -s /bin/bash <enter>
$ touch /mnt/neon/testfile <enter>
$ ls -l /mnt/neon/ <enter>
```

If you can create the testfile as shown above and see it in the directory listing, the NFS read and write access is working.
2. Create an entry in /opt/neonInsight/etc/fstab for the NFS mount on neon B1 and neon B2.  
Example:  

```
neonA.example.net:/mnt/exports /mnt/neon nfs rw,user,auto 0 0
```
3. Stop neonInsight Server on neon B1 and neon B2 by issuing the following command on each server:  

```
neonInsight stop <enter>
```
4. While in the neonInsight Server changeroot (neonInsight shell) on neon B1, copy the following directories to the NFS share:  

```
# cp -a /etc/neon /mnt/neon/ <enter>
# cp -a /var/neon/storage /mnt/neon/ <enter>
# cp -a /var/neon/local-mb /mnt/neon/ <enter>
```
5. Move /etc/neon to /etc/neon\_local and replace it with a symbolic link to the copy on the NFS share with the following commands:  

```
# mv /etc/neon /etc/neon__local <enter>
# ln -s /mnt/neon/neon /etc/neon <enter>
```
6. Repeat steps 4 and 5 on neon B2.
7. Edit neonconfig.xml on the NFS share (neon A) from either neon B1 or neon B2:  

```
# vi /mnt/neon/neon/neonconfig.xml <enter>
```

Make the following changes to the file:

Locate this line:

```
<define name="ldapserver" value="localhost"/>
```

Replace "localhost" with the IP address or hostname of neon A (*where the LDAP service is running*).

Replace all instances of "file:///var/neon/local-mb" with "file:///mnt/neon/local-mb". This change directs neonInsight server on neon B1 and neon B2 to use the NFS copy of local-mb instead of the local copy.



Replace all instances of `"/var/neon/storage/"` with `"/mnt/neon/storage/"`. This directs neonInsight Server on neon B1 and neon B2 to use the NFS copy of storage instead of the local copy. Save the file.

8. Edit `/var/www/config/conf.php` on neon B1 and neon B2 and locate the following line:  
`$conf['insight']['ldap']['server'] = 'localhost';`

Change 'localhost' above to the hostname or IP address of neon A (*where the LDAP service is running*).

9. Start neonInsight Server on neon B1 and neon B2.  
`neonInsight start <enter>`

Log into the web admin service on neon B1 and neon B2 and view the users that were added to the LDAP on neon A in step 2 (under **Configuration – neon A**).

10. Send a test message to a local user and verify that the incoming message is visible on both neon B1 and neon B2.

If an error occurs, check `/var/neon/log/logneon.log` for error messages and check the write permissions of the NFS share directories. Make sure that 'rpc.statd' and 'lockd' are running on all three servers.

## Install Balance

1. Install balance on neon A using the apt-get command:  
`apt-get install balance <enter>`
2. Start balance on neon A to test the SMTP service:  
`# balance -f smtp neonB1 neonB2`

## Testing

1. Watch the neonInsight Server logs on both neon B1 and neon B2 using the command, tail:  
`# tail -f /var/neon/log/logneon.log`
2. Test sending an email. You can use the WebClient or use the command line via telnet. To test via the command line with telnet, follow these steps:  
`# telnet neonA 25`  
Trying 192.168.81.129...  
Connected to 192.168.81.129.  
Escape character is '^]'.  
220 Neon ESMTP Server at etch.bynari.net  
SMTP Server at etch.bynari.net  
HELO test@example.net  
250 neonA.example.net  
MAIL FROM: <test@example.net>  
250 OK  
RCPT TO: <user11@bynari.net>  
250 user11 <user11@bynari.net> OK



```
DATA
354 Start mail input, end with <CRLF>.<CRLF>
From: Test User <test@example.net>
To: User 11 <user11@example.net>
Subject: Just a test.
```

This is a test message.

```
250 OK queued 0064534455-1236960468-23d6@MessageStore for
delivery, 332 bytes    ^]
telnet>
```

3. Observe the logs on neon B1 and neon B2. One of them should show that it has received and stored the test message. Repeat the test and observe the logs again. This time the other server should show activity as balance alternates between the two servers.
4. Write a script to start balance covering SMTP, IMAP, WWW, and others and deploy it on neon A:

```
balance smtp neonB1 neonB2
balance imap neonB1 neonB2
balance www neonB1 % neonB2
```

**Note:** The '%' is used as a separator for client requests. Rather than a simple round-robin algorithm, a hash distribution based on the client IP address, is used to determine which server gets the request. This feature ensures that client http sessions always route to the correct server.

With all those steps complete, any IMAP, SMTP, or WWW requests to neon A will be distributed by balance between neon B1 and neon B2. Any LDAP changes on either machine will be instantly available on the other machine as will as any incoming mail or IMAP operations.

Failure of a service on neon B1 or neon B2 will go unnoticed by the client/users except in the case of WWW. If WWW fails, the users will be asked to log in again, and a new HTTP session will be created.